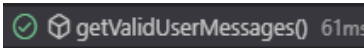
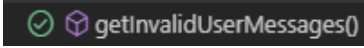
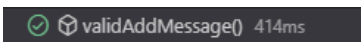
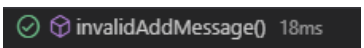









Test Result	Method being tested	Explanation
	getMessages(String user)	This is a test to check valid input on the “getMessages” method which is used to display a user's messages on the main menu page. This tests with a valid username, and checks to see if the method returns at least 1 message, since all users will have at least 1 welcome message when they create an account.
	getMessages(String user)	This is a test to check invalid input on the “getMessages” method which is used to display a user's messages on the main menu page. This tests the method with a username for an account that does not exist. It is expected that the method will return zero messages since the account does not exist. It did not return any message, thereby passing the test.
	AddMessage(String recipient, String msg, String sender)	This is a test to see if a message can be sent to a user. (adding the message to a database) The method is only supposed to send messages to users that exist in the database. So the test is done with an account that does exist. The method returns true/false depending on if the message was successfully sent. The output for the test was true, as expected, so the test passed.
	AddMessage(String recipient, String msg, String sender)	This is a test to make sure that a message can no be sent (added to the messages database) for a user that does not exist. The method returns true/false depending on if the message was successfully sent. The output for the test was false, as expected, so the test passed.
	testDatabaseConnection()	This is a test to ensure that the user has connection to the database before using DoctorsList.java or any other java files that run the SQL database

		(most of them).
 testLoginDisplay()	Login()	the JUnit test checks whether the Login window is visible or not. It creates an object of the Login class and then uses the assertTrue() method to confirm that the window is indeed visible.
 testMainMenuDisplay() 2.0s	MainMenu(String username)	This JUnit test also checks whether the main menu of an application is displayed correctly or not. It creates an object of the MainMenu class with a String parameter " user@email.com " as an example of an email and then uses the assertTrue() method to confirm that the main menu window is visible.
 testRegisterDisplay()	RegisterNewUser()	This JUnit test checks whether the RegisterNewUser window is displayed or not. It creates an object of the RegisterNewUser class and then uses the assertTrue() method to confirm that the window is visible.
 testAuthenticationValid() 470ms	authentication()	This test checks if a user with a valid email and password can log in successfully, and if so, the main menu is displayed.
 testAuthenticationInvalid() 960ms	authentication()	This test checks if the authentication method of the Login class works correctly for invalid email and password inputs. It creates a new instance of the Login class, calls the authentication method with invalid email and password, and checks if the Login window is still visible.
 testAuthenticationEmpty() 2.6s	authentication()	This test checks whether the Login page remains visible when the user attempts to authenticate with empty email and password fields. A new instance of Login is created, and the authentication method is called with empty strings for email and password. Then, it asserts that the Login page is still visible.